Performance Analysis of Information Retrieval Models on Word Pair Index Structure

N. Karthika and B. Janet

Abstract This paper analyzes the performance of word pair index structure for various information retrieval models. Word pair index structure is the most efficient for solving contextual queries and it is a precision-enhancing structure. The selection of information retrieval model is very important as it precisely influences the outcome of information retrieval system. This paper analyzes the performance of different information retrieval models using word pair index structure. It is found that there is an increase in precision of 18% when compared with traditional inverted index structure, and recall is 8% in the inverted word pair index structure.The mean average precision is increased by 26%, and R-precision is increased by 20%.

1 Introduction

Information retrieval is a process of retrieving a set of documents which are relevant to the query [1]. The user indicates his/her information need which is a description of what the user is expecting as a query. It may be a list of words or a phrase. Document collection/corpus is a group of documents. Each document contains an information that the user needs.

Indexing is a process of creating a document representation of information content which makes querying faster. There are various indexing structures available. The most popular one is inverted index structure. The Fig. 1 gives the modules of the information retrieval process using the inverted index.

Preprocessing [2] module has tokenization, stop word removal, and stemming [3]. Tokenization fragments the document into word pairs [4]. Stop word removal is an optional step which is applied to the result of tokenization where the words whose

N. Karthika (✉) · B. Janet
Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India
e-mail: bharathikarthika@gmail.com

B. Janet
e-mail: janet@nitt.edu

D. Reddy Edla et al. (eds.), *Advances in Machine Learning and Data Science*,
Advances in Intelligent Systems and Computing 705,
https://doi.org/10.1007/978-981-10-8569-7_19

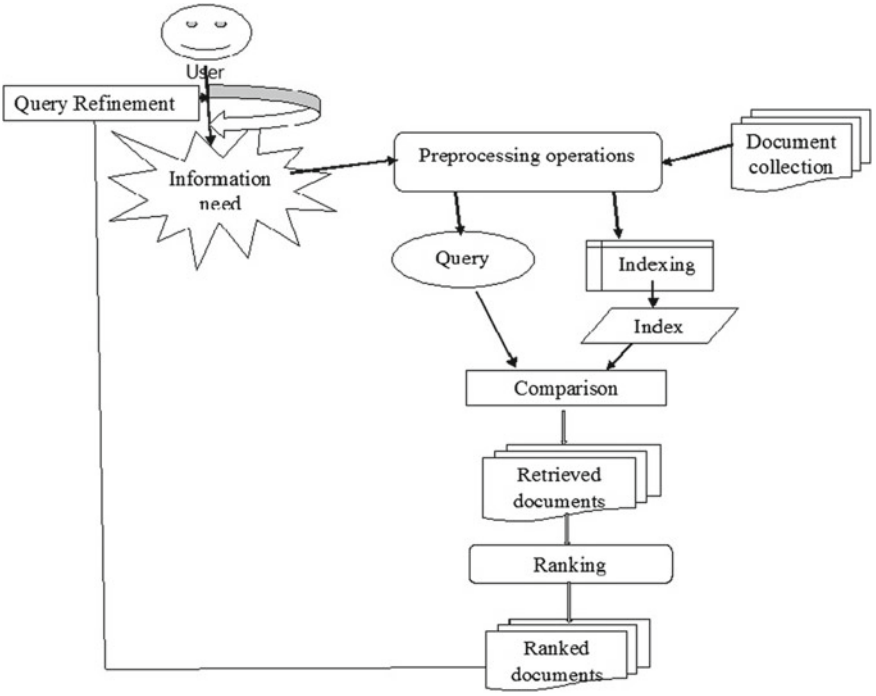

Fig. 1 Information retrieval process

contribution is very less to the meaning of the document and occurs very frequently in the document is eliminated. Then, stemming is used to normalize the word, i.e., it gives the grammatical root of word and reduces the number of words.

Comparison module fetches the documents that matches with given query. Ranking module grades the documents which are retrieved corresponding to a retrieval model based on relevance.

If the user is not satisfied with the retrieved results of documents, then he/she will refine the query accordingly to improve the retrieval performance [2].

This paper is organized as follows. The former works are summarized in Sect. 2. Various text indexing structures are explained in Sect. 3. Variety of information models are discussed in Sect. 4. The measures used in performance evaluation are described in Sect. 5. Experimental setup and results are discussed in Sect. 6. This paper analyzes the performance evaluation measures for various information retrieval models on the word pair index structure implemented with Terrier 3.5. The following are the contributions of the paper.

- For inverted word pair index structure BM25,IFB2, and TF–IDF retrieval models performed better than other retrieval models.

- It is found that there is an increase in precision of 18% when compared with traditional inverted index structure and the recall is 8% in the inverted word pair index structure.
- The mean average precision is increased by 26% and the R-precision is increased by 20%.

2 Related Works

Many statistical approaches for retrieval models have been developed, including [5–7]. According to Harter, terms are of different types [8]. They are speciality terms and non-speciality terms. Speciality terms occur frequently in documents and they contribute toward the meaning of information content. Non-speciality terms are ones whose contribution is less and they are not contributing toward the content of the documents. Both of them follow a Poisson distribution. Then it was fine-tuned by Robertson et.al and promoted by Robertson and Walker as a chain of productive implementations named BMs (eg., BM25) [9]. Later, a matured probabilistic model termed divergence-from-randomness (DFR) was used [5]. The general issues like efficiency, computational cost, and benefits of the information retrieval models were discussed in [10]. The partial and exact matching models are interdependent, and they are joined together to get better results [11]. According to the needs of the user or the application, information retrieval models have to be selected [12].

Inverted index structure is the best for evaluating Boolean queries that are ranked on huge collections. The inverted files outperform in many aspects like space, speed, and functionality in full text indexing [13]. Instead of a term indexing, phrase-based indexing achieved improved retrieval results. The phrase-based systems are precision-enhancing systems in terms of both indexing and retrieval of documents in a collection [14].

Word pair index was implemented in small document collection [15]. Word pair index was implemented on FIRE data set and found that the retrieval speed is increased [4]. This word pair index structure has not been analyzed on various retrieval models. In this paper, the analysis over various information retrieval models on word pair index structure is done using Terrier 3.5.

3 Text Indexing Structures

3.1 Inverted Index Structure

Inverted index is used in most of the search engines [13]. It has two parts. They are vocabulary and postings list. Vocabulary consists of all the distinct terms which are being indexed. Each indexed term is associated with the posting list, which has

identifier of the documents in the collection. The structure used to implement the inverted index is arrays, B-tree, B+tree, hash tables, suffix tree, and trie.

3.2 Word Pair Index Structure

Retrieval systems have queries which are of words or of a list of words [16]. In word pair indexing structure, instead of the term, every consecutive pair of terms is treated as term for representation. Compared to inverted index structure, word pair indexing contains small postings list due to the pair of words [15]. It also captures the semantic meanings of text. The queries can be resolved quicker than conventional inverted index structure since it directly captures the postings for the corresponding word pair itself.

4 Information Retrieval Models

The objective of an information retrieval process is to acquire and rank the collection of documents which are relevant to the information needs. The IR system not only finds a single document in the collection but instead identifies multiple documents with various degrees of relevance to the query. To recognize the query, information retrieval system employs a lot of models to allocate/compute a numeric score to the documents. Accordingly, it credits the ranks to the documents and retrieves the top relevant documents from the collection of documents.

4.1 Boolean Model

This model is a very classic traditional model. It is grounded in set theory and Boolean algebra [11]. This model finds the documents which are matched with the query terms exactly. Here, index term weights are considered to be binary, i.e., (0, 1). The query is framed by three logical operators (AND, OR, NOT) linked with index term. This is also known as exact match model.

It is very efficient and precise model. Due to exact matching, it retrieves either too many or very few results. Not all the queries can be easily translated into Boolean expression.

4.2 Vector Space Model

The retrieval of documents is done through partial matches unlike the Boolean match model. Compared to Boolean model, the vector space model gives better results. This model expresses queries and documents as vectors. To find out numeric score between a query and a document, this model uses cosine similarity between query vector and document vector.

$$\cos(\mathbf{q}, \mathbf{d}) = sim(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{||\mathbf{q}|| \cdot ||\mathbf{d}||} \tag{1}$$

where $\mathbf{q}$ is the query vector. $\mathbf{d}$ is the document vector. $||\mathbf{q}||$ and $||\mathbf{d}||$ are length of the query vector and document vector.

In the vector space model, computing weight of the terms that are existing in the query and the documents plays a vital role [10]. Three important elements are to be considered to compute the weight of the terms. They are term frequency t_f (frequency of the term in the document), document frequency d_f (frequency of the term in the collection), the length of the document which contains the term d_l, the number of documents n_d [17].

$$TF\_IDF = Roberston\_tf * idf * K_f \tag{2}$$

where

$\text{Roberston}\_t_f = k_1 * \left[\frac{t_f}{t_f + k_1 * \left(1 - b + \frac{b * d}{dl_{avg}} \right)} \right]$

$idf = \log\left(\frac{n_d}{d_f + 1} \right)$

d_l = The length of the document which contains the term

t_f = The frequency of the term in the document

d_f = The frequency of the term in the collection

K_f = The term frequency in the query

n_d = The number of documents

$b = 0.75 \quad k_1 = 1.2$

It produces better recall compared to the Boolean model due to the partial matching and term weighting. The ranking of documents is done with the help of relevance. Semantic sensitivity, i.e., terms with the similar contextual meaning cannot be identified.

4.3 Probabilistic Model

The probability retrieval model is established on the probability ranking principle which ranks the documents based on the probability of relevance to the query [18]. Common models are binary independence model, language models, divergence-from-randomness model [19], and Latent Dirichlet allocation [12]. It provides the partial matching with relevance ranking and queries can be expressed in easier language but the score computation is expensive.

Table 1 gives the formulae for scoring computation of different information retrieval models. The notations used in the probabilistic models are
t = The term.
d = The document in the collection.
q = The query.
t_f = Within document frequency of t in d.
N = The total number of documents in the collection.
df_t = The frequency of the term in the collection.
qt_f = The term frequency in the query.
l = The length of the document which contains the term.
l_{avg} = Average number of terms in a document.
n_t = The document frequency of t.
tfn = The normalized term frequency.
c, b, k_1 = Constants.
F = The term frequency of t in the whole collection.
λ = The variance and mean of a Poisson distribution.

5 Performance Evaluation Measures

The evaluation of an information retrieval system is the mechanism for estimating how robust or healthy a system is to meet the information needs of the users. Information retrieval systems are not only interested in quality of retrieval results (effectiveness) but also interested in quickness of results retrieved (efficiency). The methods which increase effectiveness will have a complementary effect on efficiency because of their interdependent behavior. Naturally, there is an inverse relationship between precision and recall. If the user needs to increment precision, then they have to submit small confined queries in the system, whereas if he wants to increase the recall then the expanded queries may be submitted.

The common parameters employed to compute the efficiency of the retrieval process are recall and precision.

Table 1 Probabilistic models

Retrieval model	Score computation
BM25	$w(t,d) = \sum_{t \in q \cap d} \left(\frac{tf}{tf + k_1 \cdot n_b} \cdot log\left(\frac{N - df_t + 0.5}{df_t + 0.5} \right) \cdot qt_f \right)$ $n_b = (1-b) + b \cdot \frac{l}{l_{avg}}$ $k_1 = 1.2 \quad b = 0.75$
BB2	$w(t,d) = \frac{F+1}{n_t(tfn+1)} \left(-log_2(N-1) - log_2(e) + f(N+F-1, N+F-tfn-2) - f(F, F-tfn) \right)$ $f = \frac{F}{N}$ $tfn = tf \cdot log_2\left(1 + c \cdot \frac{l_{avg}}{l}\right)$
PL2	$w(t,d) = \frac{1}{tfn+1} \left(tfn \cdot log_2 \frac{tfn}{\lambda} + \lambda + \frac{1}{tfn} - tfn \right) \cdot log_2 e + 0.5 + log_2(2\pi \cdot tfn)$ $\lambda = \frac{F}{N}$ $tfn = tf \cdot log_2\left(1 + c \cdot \frac{l_{avg}}{l}\right)$ $c = 1, ByDefault$
InL2	$w(t,d) = \frac{1}{tfn+1} \left(tfn \cdot \log_2 \frac{N+1}{n_t+0.5} \right)$ $tfn = tf \cdot log_2\left(1 + c \cdot \frac{l_{avg}}{l}\right)$ $c = 1$
IFB2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn+1)} \left(tfn \cdot \log_2 \frac{N+1}{F+0.5} \right)$ $tfn = tf \cdot log_2\left(1 + c \cdot \frac{l_{avg}}{l}\right)$ $c = 1$
In(exp)B2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn+1)} \left(tfn \cdot \log_2 \frac{N+1}{n_e+0.5} \right)$ $tfn = tf \cdot log_2\left(1 + c \cdot \frac{l_{avg}}{l}\right)$ $c = 1$
In(exp)C2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn_e+1)} \left(tfn_e \cdot \log_2 \frac{N+1}{n_e+0.5} \right)$ $n_e = N \cdot \left(1 - \left(1 - \frac{nt}{N}\right)F\right)$ $tfn_e = tf \cdot log_e\left(1 + c \cdot \frac{l_{avg}}{l}\right)$ $c = 1$
DPH	$w(t,d) = K_f * N_b * \left(tf * Idf \cdot log\left(\left(\frac{tf * l_{avg}}{l} \right) * \left(\frac{n_d}{d_f} \right) \right) + 0.5 * \text{Idf} \cdot log\left(2\pi * tf\left(1-f\right)\right) \right)$ $f = \frac{tf}{l}$ $Idf = log\left(\frac{n_d}{d_f + 1} \right)$ $N_b = \left(1-f\right) * \left(\frac{1-f}{tf+1} \right)$
Lemur TF-IDF	$w(t,d) = \left(tf_d * tf_q * idf^2 \right)$ $tf_d = \frac{k_1 * tf}{tf + k_1 * \left(1 - b + b * \frac{dl}{l_{avg}}\right)}$ $idf = log\left(\frac{n}{N} + 1 \right)$ $k_1 = 1.2 \quad b = 0.75$
DLH	$w(t,d) = \frac{1}{tf+0.5} \cdot \left(\log_2\left(\frac{tf \cdot l_{avg}}{l} \cdot \frac{N}{F}\right) + (l - tf)\log_2(1-f) + 0.5 \log_2\left(2\pi * tf(1-f)\right) \right)$ $f = \frac{tf}{l}$

5.1 Recall

Recall is the ratio of the number of relevant documents retrieved to the total number of documents relevant. It is an indicator of the exhaustivity of the indexing [1].

$$Recall = \frac{|\ relevant \quad documents \cap retrieved \quad documents\ |}{|\ relevant \quad documents\ |}$$

5.2 Precision

Precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved. It is thus an indicator of the specificity of the indexing [1].

$$Precision = \frac{|\ relevant \quad documents \cap retrieved \quad documents\ |}{|\ retrieved \quad documents\ |}$$

5.3 Mean Average Precision

In recent days, mean average precision (MAP) is highly recommended to show the quality of retrieval results in a single measure [1]. Average precision is the average of precision value acquired for the collection of top k documents existing after each relevant document is retrieved, and this is average over the queries to gain MAP. Simply MAP for a collection of queries is the mean of average precision scores for each query.

$$MAP = \frac{\sum_{q=1}^{Q} avgP(q)}{Q}$$

where Q is the total number of queries.

6 Results and Discussion

The experiment has been carried out on HP Workstation Z640 which has Intel Xeon E5-2620V3/2.4 GHz processor, 1 TB Hard Drive Capacity, 8 GB RAM for Windows 7 Professional and Java Runtime Environment of version 1.7.0.51 using Terrier 3.5. The Terrier is a suitable platform for carrying out and testing information retrieval tasks. This is an open source available at [20]. Terrier's properties are modified according to our need.

Table 2 Inverted index versus Inverted word pair index

Particulars	Inverted index	Inverted word pair index
Number of tokens	73689988	62157864
Number of terms	330737	7824526
Number of pointers	46945079	58264918
Size of index (GB)	0.22	0.99
Time to index	11 min 8 s	68 min 37 s
Retrieval time (s)	94	5
Query	50 queries	50 queries

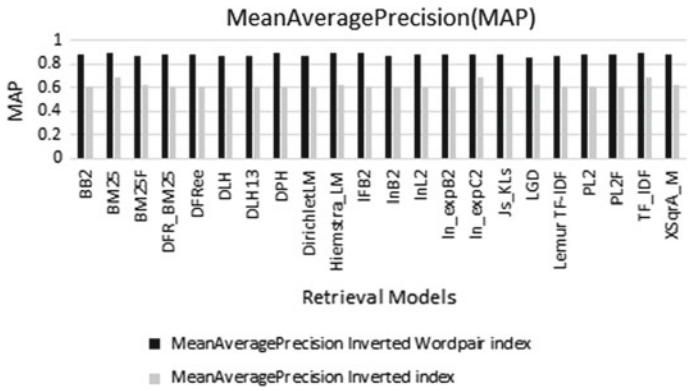

Fig. 2 Comparison of mean average precision between inverted indexing structure and inverted word pair index structure

FIRE data set has been used for testing. FIRE stands Forum for Information Retrieval and Evaluation. FIRE collection maintains the same classic representation of TREC collection. The FIRE 2011 comprises of various documents from newspapers and Websites. It is available and has been downloaded from [21]. Table. 2 shows the comparison between traditional inverted index and inverted word pair index structure for FIRE data set with time to index, retrieval time, and query size.

Figure 2 depicts the mean average precision (MAP) between inverted index structure and inverted word pair index structure.The modified inverted word pair index structure gave better precision value than traditional inverted index structure.The MAP is increased by 26% in inverted word pair index structure. For inverted word pair index structure, BM25,IFB2, and TF–IDF retrieval models performed better than other retrieval models.

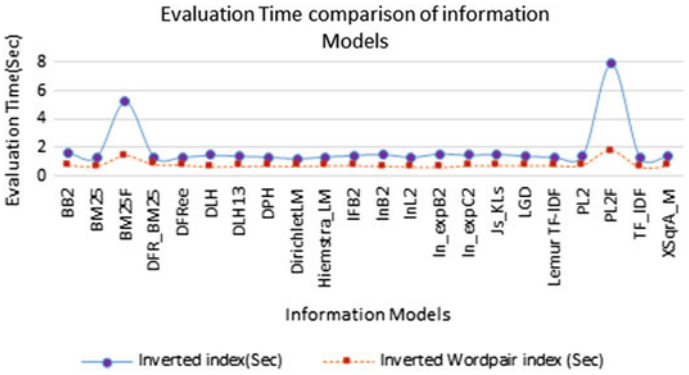

Fig. 3 Evaluation time of various information models

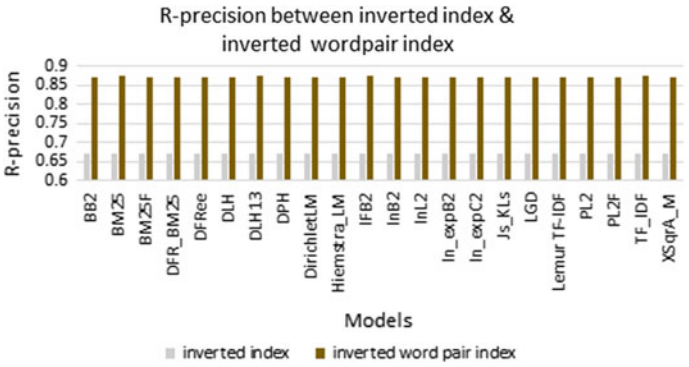

Fig. 4 R-precision comparison of inverted index and inverted word pair indexing structure

Figure 3 shows the time taken by various information models to evaluate the performance. In both the structures, PL2F model takes more time than other models. BM25F takes second largest time than others.

Figure 4 illustrates the R-precision comparison of inverted index and inverted word pair index structure. For inverted index structure, BM25 and IFB2 information retrieval models perform better than others. For inverted word pair index structure, BM25, DLH13, and IFB2 models gave better results than others.

Figures 5 and 6 show the precision–recall curve for BM25 and TF–IDF information retrieval models. The precision–recall curve for inverted word pair index structure appears smoother than the inverted index structure.

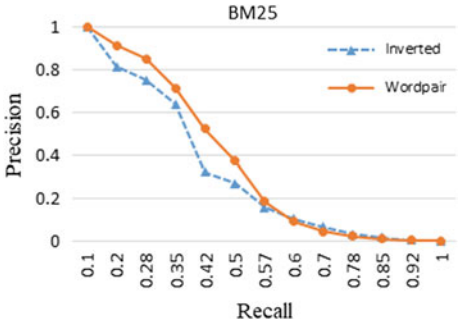

Fig. 5 Precision–recall curve for BM25 Model

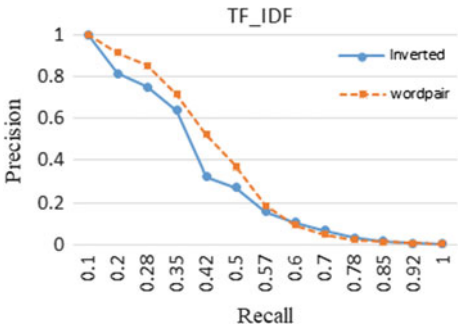

Fig. 6 Precision–recall curve for TF–IDF Model

7 Conclusion

From the results, we come to a conclusion that for inverted word pair index structure BM25,IFB2, and TF–IDF retrieval models performed better than other retrieval models. In the same way, for traditional inverted index structure BM25, In-expC2, TF–IDF retrieval models have better performance than others. It is found that there is an increase in precision of 18% when compared with traditional inverted index structure, and recall is reduced by 8% in the inverted word pair index structure.The mean average precision is increased by 26%, and R-precision is increased by 20%. In future, it can be implemented on big data and the results can be analyzed for various applications.

References

1. Salton, G., McGill, M.J.: Introduction to modern information retrieval. In: McGraw-Hill Computer Science Series. McGraw-Hill (1983)
2. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
3. Porter, M.F.: Readings in information retrieval. In: Chapter An Algorithm for Suffix Stripping, pp. 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997)

4. Karthika, N., Janet, B.: Word pair index structure for information retrieval using terrier 3.5. In: IEEE Technically Sponsored International Conference on Computational Intelligence on Data Science (ICCIDS) June, 2017 (In Press)
5. Amati, Gianni, Rijsbergen, Cornelis Joost Van: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. (TOIS) **20**(4), 357–389 (2002)
6. Maron, M.E., Kuhns, J.L.: On relevance, probabilistic indexing and information retrieval. J. ACM (JACM), **7**(3), 216–244 (1960)
7. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. J. Assoc. Inf. Sci. Technol. **27**(3), 129–146 (1976)
8. Harter, S.P.: A probabilistic approach to automatic keyword indexing. Part ii. An algorithm for probabilistic indexing. J. Assoc. Inf. Sci. Technol. **26**(5), 280–289 (1975)
9. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 232–241. Springer, New York, Inc. (1994)
10. Dong, H., Hussain, F.K., Chang, E.: A survey in traditional information retrieval models. In: 2008 2nd IEEE International Conference on Digital Ecosystems and Technologies, DEST 2008, pp. 397–402. IEEE (2008)
11. Ruban, S., Sam, S.B., Serrao, L.V., Harshitha.: A Study and Analysis of Information Retrieval Models, pp. 230–236 (2015)
12. Jiang, H.: Study on the performance measure of information retrieval models. In: 2009 International Symposium on Intelligent Ubiquitous Computing and Education, pp. 436–439. IEEE (2009)
13. Zobel, Justin, Moffat, Alistair, Ramamohanarao, Kotagiri: Inverted files versus signature files for text indexing. ACM Trans. Database Syst. (TODS) **23**(4), 453–490 (1998)
14. Mitra, Mandar, Chaudhuri, B.B.: Information retrieval from documents: a survey. Inf. Retr. **2**(2–3), 141–163 (2000)
15. Janet, B., Reddy, A.V.: Wordpair index: a nextword index structure for phrase retrieval. Int. J. Recent Trends Eng. Technol. **3**(2) (2010)
16. Bahle, D., Williams, H.E., Zobel. J.: Efficient phrase querying with an auxiliary index. In: Proceedings of the 25th Annual International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 215–221. ACM (2002)
17. Paik, J.H.: A novel tf-idf weighting scheme for effective ranking. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 343–352. ACM (2013)
18. Gasmi, K., Khemakhem, M.T., Jemaa, M.B.: Word indexing versus conceptual indexing in medical image retrieval. In: CLEF (Online Working Notes/Labs/Workshop) (2012)
19. Singh, A., Dey, N., Ashour, A.S., Santhi, V.: Web Semantics for Textual and Visual Information Retrieval, 01 (2017)
20. Retrieval platform. http://ir.dcs.gla.ac.uk/terrier/
21. Fire dataset. http://storm.cis.fordham.edu/~gweiss/data-mining/datasets.html